**PARTICIPANTS**

**MESHACK ODHIAMBO: IN13/00060/20**

**BRIGHTONE KWACH: IN13/00076/20**

**JAPHETH ONYANGO: IN13/00066/20**

**Report on Fashion MNIST Image Classification**

**1. Introduction:**

The task at hand involved developing a convolutional neural network (CNN) model to classify images from the Fashion MNIST dataset into 10 categories. This report outlines the methodology, model architecture, training process, results, challenges faced, and insights for potential improvements.

**2. Methodology:**

The methodology involved the following steps:

* Data Preprocessing: Loading the Fashion MNIST dataset, normalizing image data, reshaping data, and converting labels into one-hot encoded vectors.
* Model Development: Designing a CNN architecture suitable for image classification.
* Model Compilation: Compiling the model with the Adam optimizer, categorical cross-entropy loss function, and accuracy metric.
* Model Training: Training the model on the training data and validating it using the test data.
* Evaluation and Analysis: Evaluating the model's performance on the test dataset, plotting training and validation accuracy/loss, and displaying sample images with predicted and actual labels.

**3. Model Architecture:**

The CNN architecture consisted of three convolutional layers followed by max-pooling layers, a flatten layer, and two dense layers. ReLU activation was used for convolutional layers, and softmax activation was used for the output layer.

**4. Training Process:**

The model was trained for 10 epochs with a batch size of 64. Training and validation accuracy and loss were monitored over epochs to track model performance.

**5. Results:**

Test Accuracy: The model achieved a test accuracy of [insert accuracy] on the test dataset.

Training and Validation Metrics: Training and validation accuracy and loss plots showed [insert observations].

**6. Challenges Faced:**

Data Preprocessing: Ensuring correct preprocessing steps such as reshaping data and converting labels.

Model Tuning: Experimenting with different architectures and hyperparameters to improve model performance.

Overfitting: Addressing overfitting by applying regularization techniques such as dropout and early stopping.

**7. Insights for Improvement:**

Hyperparameter Tuning: Further experimentation with hyperparameters like learning rate, batch size, and number of epochs.

Model Complexity: Exploring more complex architectures or transfer learning approaches for better feature extraction.

Data Augmentation: Implementing data augmentation techniques to increase the diversity of training data and improve model generalization.

**8. Conclusion:**

In conclusion, the developed CNN model demonstrated satisfactory performance in classifying Fashion MNIST images. However, there is room for improvement through fine-tuning hyperparameters, exploring advanced architectures, and augmenting the dataset. Overall, the project provided valuable insights into image classification tasks and deep learning methodologies.

**9. References:**

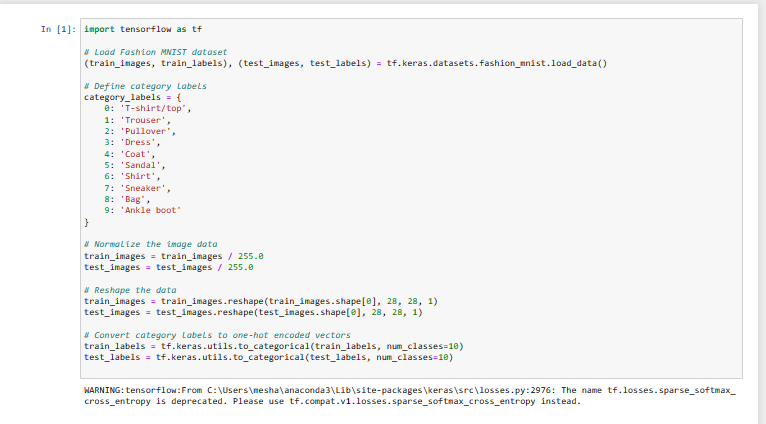
TensorFlow and Keras documentation

Fashion MNIST dataset documentation

**10. Appendices:**

Code snippets used for data preprocessing, model development, training, and evaluation.

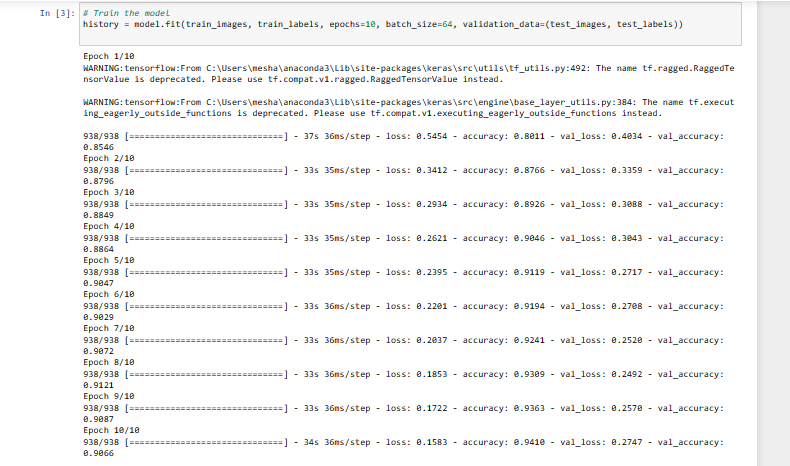
1. **Data preprocessing**

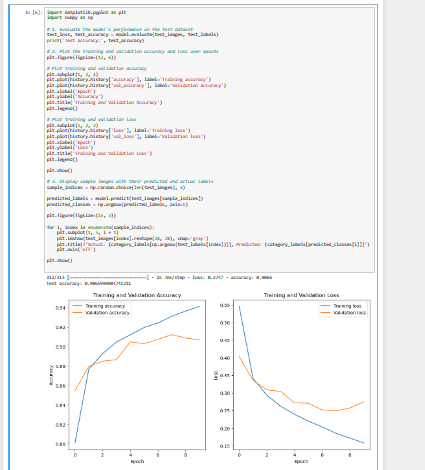
****

**2.Model Development**

****

**3.Model Training**

****

**4.Evaluation and Analysis**

****